Solving traveling salesman problem via clustering and a new algorithm for merging tours

V.A. TRAPEZNIKOV INSTITUTE OF CONTROL SCIENCES OF RUSSIAN ACADEMY OF SCIENCES

Egor Barashov*, Nikita Shushko, Simon Krasotkin, Daria Lemtuzhnikova[1]

1.Trapeznikov Institute of Control Sciences of RAS, Moscow, Russia

Motivation

In solving real-world problems, the number of vertices can be so large that employing algorithms for exact solutions is impractical. In such cases, a decomposition approach may be employed, involving:

1. Clustering: identifying subproblems of smaller dimensions;
2. Solving: solving each subproblem by finding a cycle within a cluster;
3. Merging: forming the final solution by merging the solutions of the subproblems.

In our work:

- we propose a new merging algorithm based on the pairwise merging of clusters
- compare this merging algorithm with other;
- analysis of the errors associated with this approach, evaluating the impact of errors at each stage on the overall solution error.

We use

Clustering algorithms:

- k-means,
- affinity propagation.

Google OR-Tools

We solve each of these subproblems separately using Google's 'or-tools' library.

Our algorithm

Given k cycles C1, C2, . . . , Ck, the algorithm repeatedly selects two neighboring clusters to merge via clusters distances. or two cycles Cp and Cq we aim to merge them by identifying the best pair of edges from each cycle to replace. To do this, we solve the following problem:

$$\min \left(d(i_t^q, i_s^p) + d(i_{t+1}^q, i_{s+1}^p) - d(i_t^q, i_{t+1}^q) - d(i_s^p, i_{s+1}^p)\right),$$

where d(i, j) is the distance between cities i and j.
This process of merging two cycles is repeated until all clusters are combined into one final cycle.

Mid-edge merging algorithm

Given two cycles Cp and Cq, we first calculate the midpoints of each edge:

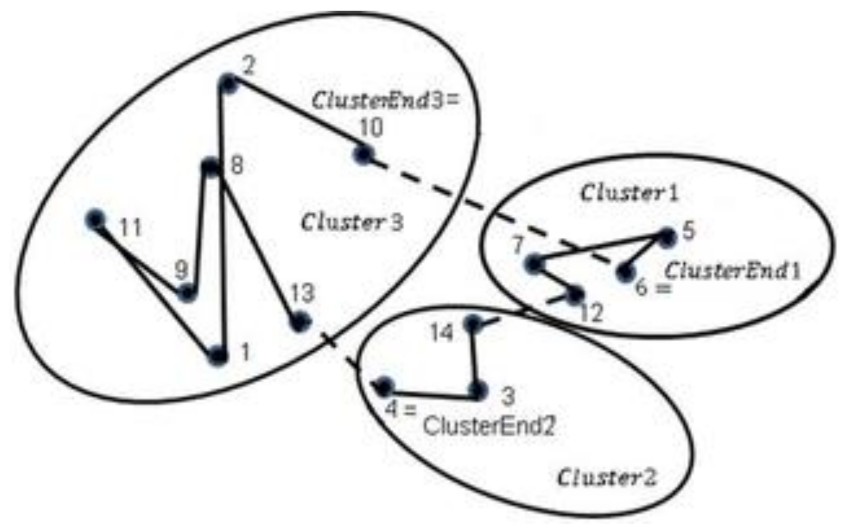
$$m_s^p = \frac{i_s^p + i_{s+1}^p}{2}, \quad m_t^q = \frac{i_t^q + i_{t+1}^q}{2}$$

The algorithm identifies the pair of edges (one from each cluster) whose midpoints are closest by minimizing the distance between midpoints:

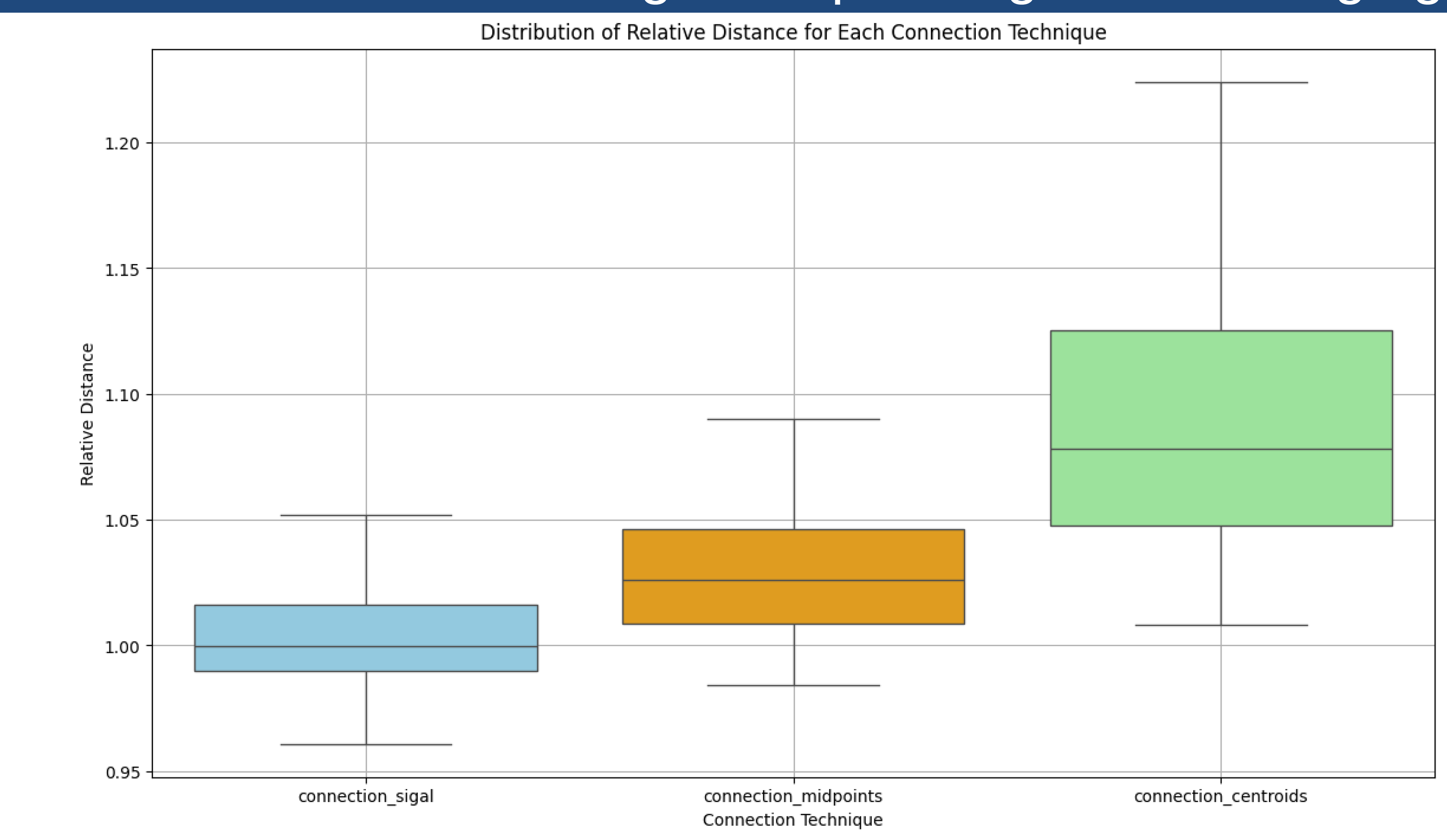
$$\min d(m_s^p, m_t^q).$$

Centroid-based merging algorithm

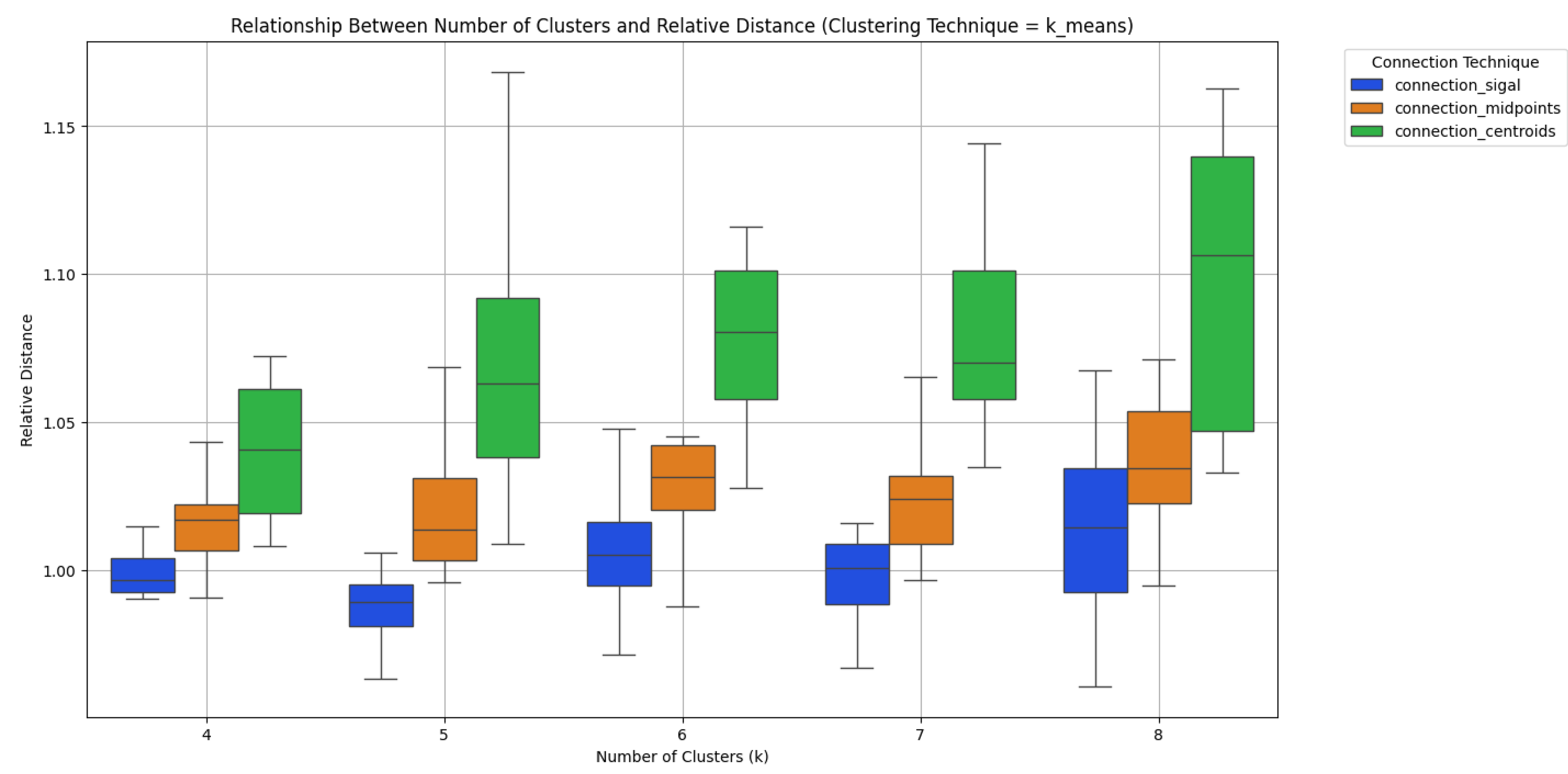
This algorithm focuses on merging clusters based on the distances between their centroids and the nearest cities within each cluster. A centroid represents the geometric center of a cluster.



Variation in solution lengths depending on the merging method



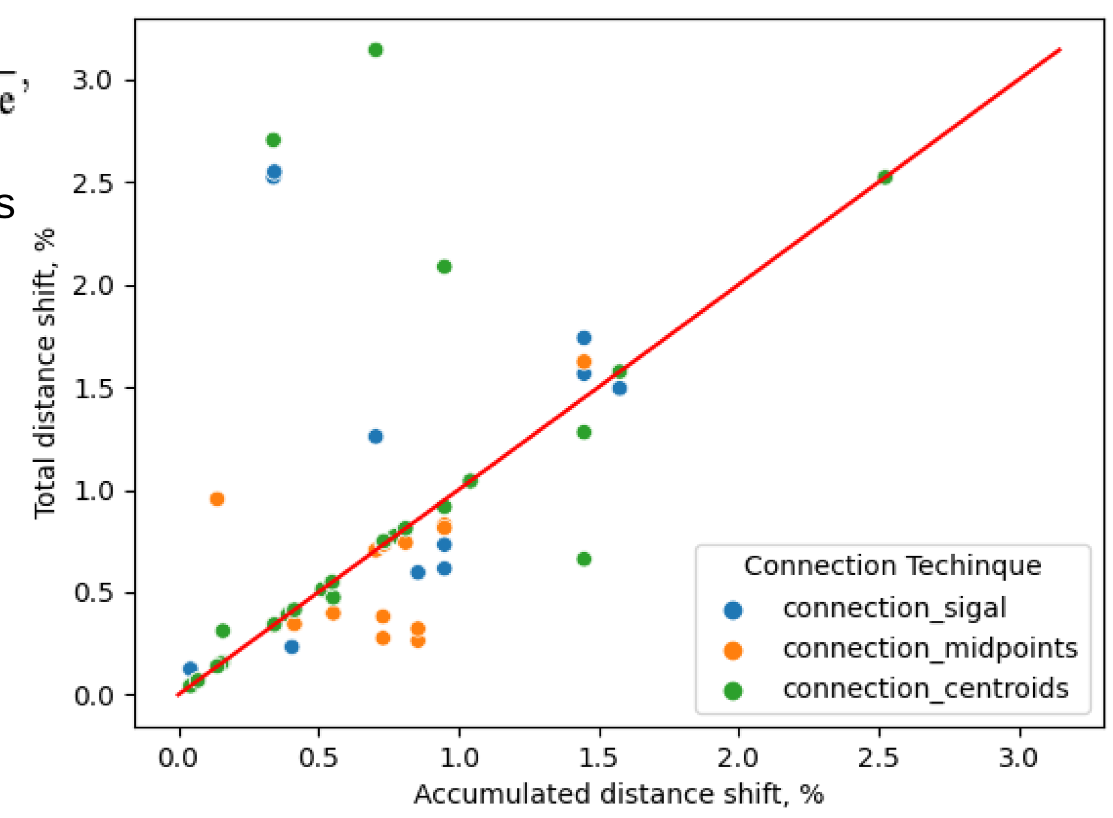
The relationship between the cycle length and the number of clusters



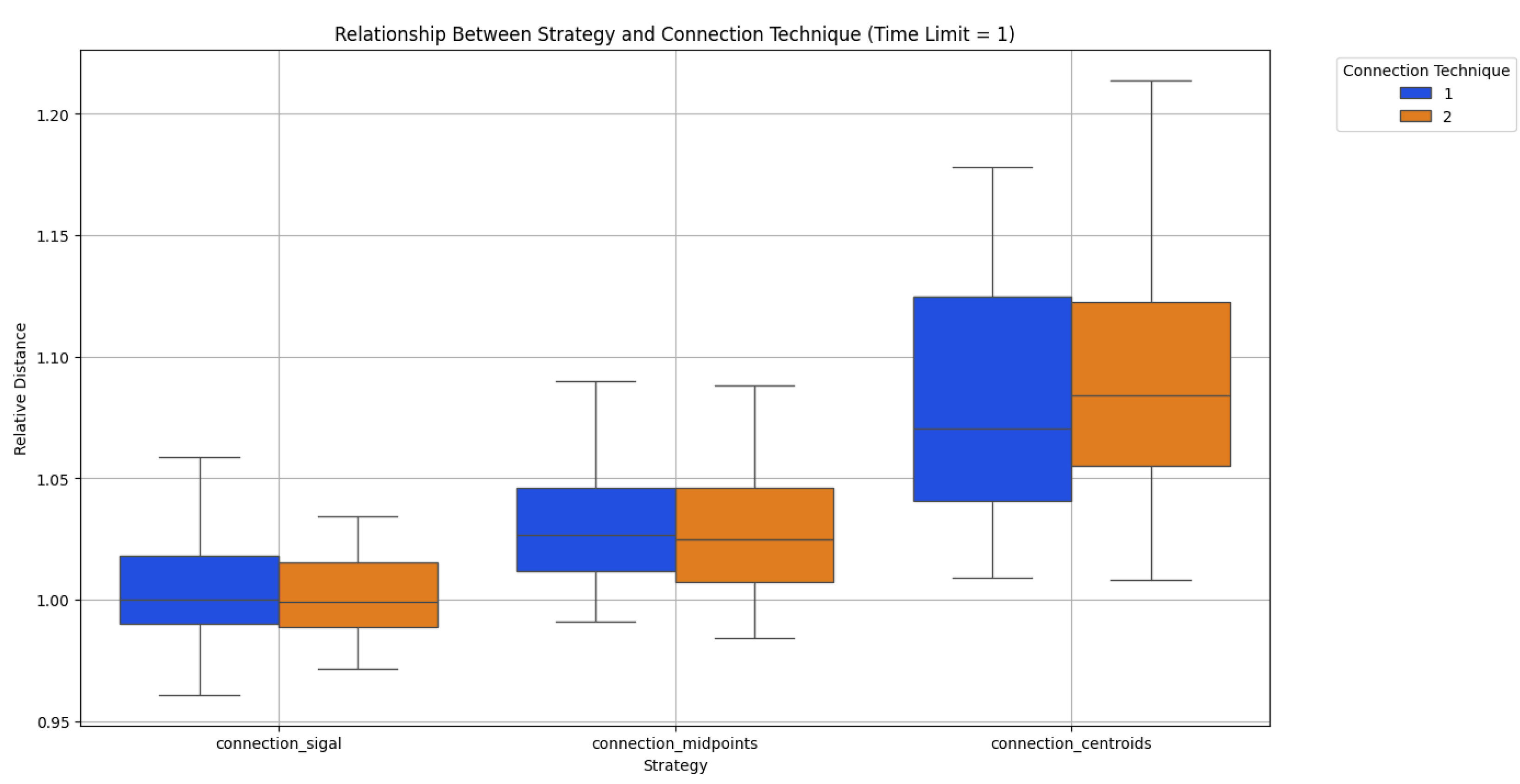
Relationship between variation in accumulated distance and total distance after time limit changes

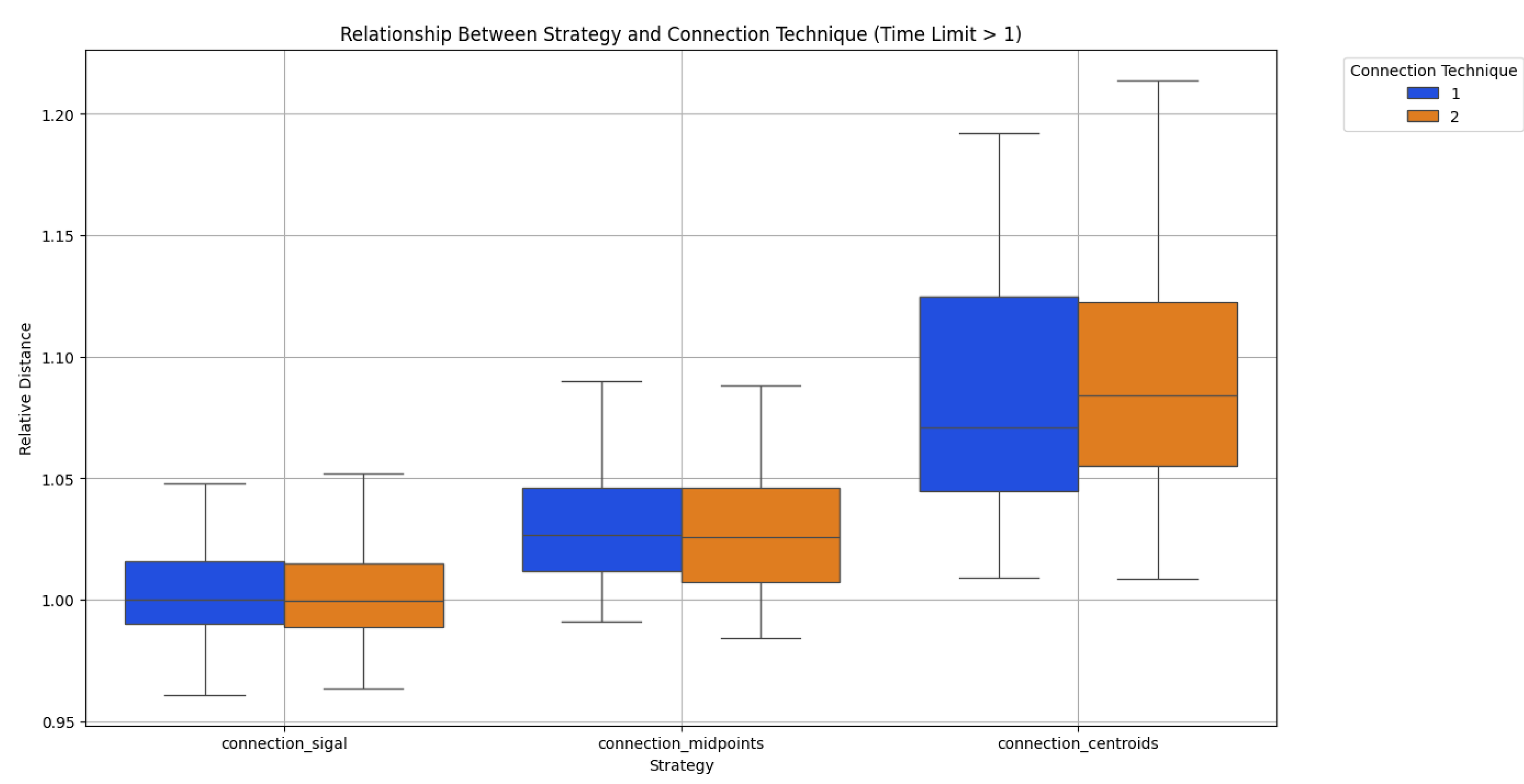
$$\text{Relative Distance} = \frac{\text{Total Distance}}{\text{Accumulated Distance}},$$

where the Accumulated Distance is the sum of the solutions for all clusters.



The dependence of the relative solution length on the merging method and cluster connection order strategy for two time limits of 1 and 10 seconds





ICOMP INTERNATIONAL CONFERENCE ON COMPUTATIONAL OPTIMIZATION

Corresponding author:
barashov.eb17@physics.msu.ru